

Feature Reduction via Generalized Uncorrelated Linear Discriminant Analysis

Jieping Ye^{*} Ravi Janardan[†] Qi Li[‡] Haesun Park[§]

Abstract

High-dimensional data appear in many applications of data mining, machine learning, and bioinformatics. Feature reduction is commonly applied as a preprocessing step to overcome the curse of dimensionality. Uncorrelated Linear Discriminant Analysis (ULDA) was recently proposed for feature reduction. The extracted features via ULDA were shown to be statistically uncorrelated, which is desirable for many applications.

In this paper, an algorithm called ULDA/QR is proposed to simplify the previous implementation of ULDA. Then the ULDA/GSVD algorithm is proposed based on a novel optimization criterion, to address the singularity problem which occurs in undersampled problems, where the data dimension is larger than the data size. The criterion used is the regularized version of the one in ULDA/QR. Surprisingly, our theoretical result shows that the solution to ULDA/GSVD is independent of the value of the regularization parameter. Experimental results on various types of datasets are

^{*}Department of Computer Science & Engineering, Arizona State University, Tempe, AZ 85281, U.S.A. jieping.ye@asu.edu.

[†]Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN 55455, U.S.A. janardan@cs.umn.edu.

[‡]Department of Computer Science, University of Delaware, Newark, DE 19716, USA. qili@cis.udel.edu.

[§]College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A., and CISE/CCF, the National Science Foundation, 4201 Wilson Boulevard, Arlington, Virginia 22230, U.S.A. hpark@cc.gatech.edu.

reported to show the effectiveness of the proposed algorithm and to compare it with other commonly used feature reduction algorithms.

Keywords: Feature reduction, Uncorrelated Linear Discriminant Analysis, QR-decomposition, Generalized Singular Value Decomposition.

1 Introduction

Feature reduction is important in many applications of data mining, machine learning, and bioinformatics, because of the so-called curse of dimensionality [6, 10, 14]. Many methods have been proposed for feature reduction, such as Principal Component Analysis (PCA) [19], Linear Discriminant Analysis (LDA) [10], etc. LDA aims to find optimal discriminant features by maximizing the ratio of the between-class distance to the within-class distance of a given dataset under supervised learning conditions. It has been successfully employed in many applications including information retrieval [2, 4], face recognition [1, 27, 28] and microarray data analysis [7]. Its simplest implementation, so-called *classical LDA*, applies an eigen-decomposition on the scatter matrices, but fails when the scatter matrices are singular, as is the case for undersampled data. This is known as the *singularity problem* or *undersampled problem* [20].

Uncorrelated features¹, are desirable in many applications, because they contain minimum redundancy. Motivated by extracting feature vectors having uncorrelated features, *uncorrelated LDA* (ULDA), was recently proposed in [17, 18]. However, the proposed algorithm in [17] involves a sequence of generalized eigenvalue problems. It is computationally expensive for large and high-dimensional datasets. Like classical LDA, it does not address the singularity problem either. We

¹Two variable x and y are said to be uncorrelated, if their covariance is zero, i.e., $\text{cov}(x, y) = 0$

thus call it *classical ULDA*. More details can be found in Section 3.

Classical LDA and classical ULDA were introduced from different perspectives. But it has been found that there is relationship between classical LDA and classical ULDA [18]. More precisely, under the assumption that the eigenvalue problem in classical LDA has no multiple eigenvalues, it was shown that classical ULDA is equivalent to classical LDA [18]. In this paper, we will show that the equivalence between these two still holds without the above assumption. Based on this equivalence, ULDA/QR is proposed to simplify the ULDA implementation in [17]. Here ULDA/QR denotes ULDA based on QR-decomposition.

Classical LDA and classical ULDA do not address the singularity problem, hence it is difficult to apply them to undersampled data. Such high-dimensional, undersampled problems frequently occur in many applications including information retrieval [15], face recognition [27] and microarray analysis [7]. Several schemes have been proposed to address the singularity problem in classical LDA in the past, including pseudo-inverse based LDA [30], the subspace based method [27], regularization [9], and the method based on the generalize singular value decomposition, called LDA/GSVD [23, 15, 16], etc. Pseudo-inverse based LDA applies pseudo-inverse [11] to deal with the singularity problem. The subspace-based method applies the Karhunen-Loeve (KL) expansion, also known as Principal Component Analysis (PCA) [19], before LDA. Its limitation is that some useful information may be lost in the KL expansion. Regularized LDA overcomes the singularity problem by increasing the magnitude of the diagonal elements of the scatter matrices (usually by adding a scaled identity matrix). The difficulty in using regularized LDA for feature reduction is the choice of the amount of perturbation. A small perturbation is desirable to preserve the original matrix structure, while a large perturbation is more effective in dealing

with the singularity problem.

There is much less work on addressing the singularity problem in classical ULDA than on classical LDA. In the subspace ULDA presented in [17], a subspace based method was applied (PCA is applied to the between-class scatter matrix).

We address the singularity problem in ULDA, in the second part of this paper, by introducing a novel optimization criterion that combines the key ingredients of ULDA/QR and regularized LDA. The criterion is the perturbed version of the criterion used in ULDA/QR. Based on this criterion and the Generalized Singular Value Decomposition (GSVD) [21], we propose a novel feature reduction algorithm, called ULDA/GSVD. ULDA/GSVD solves the singularity problem directly, thus avoiding the information loss that occurs in the subspace method. The difference between ULDA/GSVD and the traditional regularized LDA is that the optimal discriminant feature vectors via ULDA/GSVD are independent of the value of regularization parameter. This is quite a surprising result and the proof and the details are given in Section 5.

With the K-Nearest-Neighbor (K-NN) classifier, we evaluate the effectiveness of ULDA/GSVD and compare it with several other commonly used feature reduction algorithms, including Orthogonal Centroid Method (OCM) [24], PCA [19], and subspace ULDA [17], on various types of datasets, including text documents, face images, microarray gene expression data, etc. The experimental results show that the ULDA/GSVD algorithm is competitive with other existing algorithms including PCA, OCM, and subspace ULDA. Results also show that ULDA/GSVD is stable under different K-NN classifiers.

The rest of the paper is organized as follows: Sections 2 and 3 give brief reviews on classical LDA and classical ULDA, respectively. The ULDA/QR algorithm is presented in Section 4. Section 5

Table 1. Summary of notations used

Notations	Descriptions	Notations	Descriptions
A	data matrix	n	number of training data points
N	dimension of the training data	ℓ	reduced dimension
k	number of classes	S_b	between-class scatter matrix
S_w	within-class scatter matrix	S	total scatter matrix
G	transformation matrix	K	number of nearest neighbors in K-NN
A_i	data matrix of the i -th class	S_i	covariance matrix of the i -th class
c_i	mean of data in the i -th class	P_i	<i>a priori</i> probability of the i -th class
c	total mean of the training data	t	rank of the matrix S

proposes the ULDA/GSVD algorithm, based on a novel criterion that is the regularized version of the criterion used in ULDA/QR. We prove theoretically that the solution to ULDA/GSVD is independent of the value of regularization applied. Experimental results are presented in Section 6. We conclude in Section 7. For convenience, the important notations used in this paper are listed in Table 1.

2 Classical Linear Discriminant Analysis

Given a data matrix $A = (a_{ij}) \in \mathbb{R}^{N \times n}$, where each column corresponds to a data point and each row corresponds to a particular feature, we consider finding a linear transformation $G \in \mathbb{R}^{N \times \ell}$ ($\ell < N$) that maps each column a_i , for $1 \leq i \leq n$, of A in the N -dimensional space to a vector y_i in the ℓ -dimensional space as follows:

$$G : a_i \in \mathbb{R}^N \rightarrow y_i = G^T a_i \in \mathbb{R}^\ell.$$

The resulting data matrix $Z = G^T A \in \mathbb{R}^{\ell \times n}$ contains ℓ rows, i.e. there are ℓ features for each data point in the dimension reduced (transformed) space. It is also clear that the features in the

dimension reduced space are linear combinations of the features in the original high dimensional space, where the coefficients of the linear combinations depend on the transformation matrix G .

A common way to compute the transformation matrix G , for clustered data sets, is through classical LDA. Classical LDA computes the optimal transformation matrix G such that the class structure is preserved. More details are given below.

Assume that there are k classes in the dataset. Suppose c_i , S_i , P_i are the mean vector, covariance matrix, and *a priori* probability of the i -th class, respectively, and c is the total mean. Then the *between-class scatter matrix* S_b , the *within-class scatter matrix* S_w , and the *total scatter matrix* S are defined as follows [10]:

$$\begin{aligned} S_w &= \sum_{i=1}^k P_i S_i, \\ S_b &= \sum_{i=1}^k P_i (c_i - c)(c_i - c)^T, \\ S &= S_b + S_w. \end{aligned}$$

The covariance matrix S_i of the i th class can be decomposed as

$$S_i = A_i A_i^T,$$

where each column in A_i corresponds to a data point from the i -th class subtracted by its mean m_i .

Define the matrices [15, 16]

$$H_w = [\sqrt{P_1}A_1, \dots, \sqrt{P_k}A_k], \quad (1)$$

$$H_b = [\sqrt{P_1}(c_1 - c), \dots, \sqrt{P_k}(c_k - c)]. \quad (2)$$

$$H_t = A - ce^T, \quad (3)$$

where $e \in \mathbb{R}^n$ is a vector of ones.

Then the scatter matrices S_w , S_b , and S can be expressed as

$$S_w = H_w H_w^T, \quad S_b = H_b H_b^T, \quad S = H_t H_t^T.$$

The *traces* of the two scatter matrices can be computed as follows:

$$\begin{aligned} \text{trace}(S_w) &= \sum_{i=1}^k P_i \|A_i\|_F^2, \\ \text{trace}(S_b) &= \sum_{i=1}^k P_i \|c_i - c\|^2, \end{aligned}$$

where $\|\cdot\|_F$ denotes the Frobenius norm [11]. Hence, $\text{trace}(S_w)$ measures the closeness of the vectors within classes, while $\text{trace}(S_b)$ measures the separation between classes.

In the lower-dimensional space resulting from the linear transformation G , the within-class scatter and between-class scatter matrices become

$$\begin{aligned} S_w^L &= (G^T H_w)(G^T H_w)^T = G^T S_w G, \\ S_b^L &= (G^T H_b)(G^T H_b)^T = G^T S_b G. \end{aligned}$$

An optimal transformation G would maximize $\text{trace}(S_b^L)$ and minimize $\text{trace}(S_w^L)$. Classical LDA aims to compute the optimal G , such that

$$G = \arg \max_G \text{trace} \left((G^T S_w G)^{-1} G^T S_b G \right). \quad (4)$$

Other optimization criteria, including those based on the determinant could also be used instead [6, 10]. The solution to the optimization problem in Eq. (4) can be obtained by solving an eigenvalue problem on $S_w^{-1} S_b$ [10], provided that the within-class scatter matrix S_w is nonsingular.

Since the rank of the between-class scatter matrix is bounded from above by $k - 1$, there are at

most $k - 1$ discriminant vectors by classical LDA. A stable way to solve this eigenvalue problem is to apply SVD on the scatter matrices. Details can be found in [27].

Classical LDA is equivalent to maximum likelihood classification assuming normal distribution for each class with the common covariance matrix. Although relying on assumptions which do not hold in many applications, LDA has been proven to be effective. This is mainly due to the fact that a simple, linear model is more robust against noise, and most likely will not overfit. Generalization of LDA by fitting Gaussian mixtures to each class has been studied in [13].

Classical LDA can not handle singular scatter matrices, which limits its applicability to low-dimensional data. Several methods, including pseudo-inverse based LDA [30], subspace LDA [27], regularized LDA [9], LDA/GSVD [23, 15, 16], and Penalized LDA [12], were proposed in the past to deal with the singularity problem. More details can be found in [22, 29].

In pseudo-inverse based LDA, pseudo-inverse is applied to avoid the singularity problem, which is equivalent to approximating the solution using a least-square method. In subspace LDA, an intermediate dimension reduction algorithm, such as PCA, is applied to reduce the dimension of the original data, before classical LDA is applied. A limitation of this approach is that the optimal value of the reduced dimension for the intermediate dimension reduction algorithm is difficult to determine. In regularized LDA, a positive constant μ is added to the diagonal elements of S_w , as $S_w + \mu I_N$, where I_N is an identity matrix. The matrix $S_w + \mu I_N$ is positive definite, for any $\mu > 0$, hence nonsingular. A limitation of this approach is that the optimal value of the parameter μ is difficult to determine. Cross-validation is commonly applied to estimate the optimal μ . More details can be found in [20].

3 Uncorrelated Linear Discriminant Analysis (ULDA)

ULDA aims to find the optimal discriminant vectors that are S -orthogonal². Specifically, suppose r vectors $\phi_1, \phi_2, \dots, \phi_r$ are obtained, then the $(r+1)$ -th vector ϕ_{r+1} is found to maximize the Fisher criterion function [17]

$$f(\phi) = \frac{\phi^T S_b \phi}{\phi^T S_w \phi},$$

subject to the constraints:

$$\phi_{r+1}^T S \phi_i = 0, \quad (i = 1, \dots, r).$$

The algorithm in [17] finds ϕ_i successively as follows: The j -th discriminant vector ϕ_j of ULDA is the eigenvector corresponding to the maximum eigenvalue of the following generalized eigenvalue problem:

$$U_j S_b \phi_j = \lambda_j S_w \phi_j,$$

where

$$U_1 = I_N,$$

$$D_j = [\phi_1, \dots, \phi_{j-1}]^T \quad (j > 1),$$

$$U_j = I_N - S D_j^T (D_j S S_w^{-1} S D_j^T)^{-1} D_j S S_w^{-1} \quad (j > 1),$$

and I_N is the identity matrix.

Assume that $\{\phi_i\}_{i=1}^d$ are the d optimal discriminant vectors for the above ULDA formulation.

Then the original data matrix A is transformed into $Z = G^T A$, where $G = [\phi_1, \dots, \phi_d]$. The i -th

²Two vectors x and y are S -orthogonal, if $x^T S y = 0$.

feature component of Z is $z_i = \phi_i^T A$, and the covariance between z_i and z_j is

$$\begin{aligned}
\text{Cov}(z_i, z_j) &= E(z_i - Ez_i)(z_j - Ez_j) \\
&= \phi_i^T \{E(A - EA)(A - EA)^T\} \phi_j \\
&= \phi_i^T S \phi_j.
\end{aligned} \tag{5}$$

Hence, their correlation coefficient is

$$\text{Cor}(Z_i, Z_j) = \frac{\phi_i^T S \phi_j}{\sqrt{\phi_i^T S \phi_i} \sqrt{\phi_j^T S \phi_j}}. \tag{6}$$

Since the discriminant vectors of ULDA are S -orthogonal, i.e. $\phi_i^T S \phi_j = 0$, for $i \neq j$, we have $\text{Cor}(Z_i, Z_j) = 0$, for $i \neq j$. That is, the feature vectors transformed by ULDA are mutually uncorrelated. This is a desirable property for feature reduction. More details on the role of uncorrelated attributes can be found in [17]. The limitation of the above ULDA algorithm is the expensive computation of the d generalized eigenvalue problems, where d is number of optimal discriminant vectors by ULDA.

In the literature for LDA, Foley-Sammon Linear Discriminant Analysis (FSLDA), which was proposed by Foley and Sammon for two-class problems [8], has also received attention. It was then extended to the multi-class problems by Duchene and Leclercq [5]. Both ULDA and FSLDA use the same Fisher criterion function. The main difference is that the optimal discriminant vectors generated by ULDA are S -orthogonal to each other, while the optimal discriminant vectors by FSLDA are orthogonal to each other.

4 The ULDA/QR algorithm

In this section, we first show the equivalence between classical ULDA and a variant of classical LDA, which holds regardless of the distribution of the eigenvalues of $S_w^{-1}S_b$. This result enhances the one in [18] where the equivalence between these two is based on the assumption that there are no multiple eigenvalues for $S_w^{-1}S_b$ (note that both results assume that the within-class scatter matrix S_w is nonsingular). Based on the equivalence, we propose ULDA/QR to simplify the ULDA implementation in [17].

Consider a variant of classical LDA in Eq. (4) as follows:

$$G = \arg \max_{G^T S G = I_\ell} F(G), \quad (7)$$

where

$$F(G) = \text{trace} \left((G^T S_w G)^{-1} G^T S_b G \right). \quad (8)$$

The use of the total scatter S in discriminant analysis has been discussed in [3]. Note that the ULDA algorithm discussed in the previous section finds the discriminant vectors in G successively. However in the new formulation above, we compute all discriminant vectors simultaneously. The S -orthogonality constraint is enforced as a constraint. Our main result in this section, summarized in Theorem 4.2, shows that these two formulations for ULDA are equivalent.

The main technique for solving the optimization problem in Eq. (7) is the simultaneous diagonalization of the within-class and between-class scatter matrices. It is well known that for a symmetric positive definite matrix S_w and a symmetric matrix S_b , there exists a nonsingular

matrix X such that

$$X^T S_w X = I_N, \quad (9)$$

$$X^T S_b X = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_N), \quad (10)$$

where $\lambda_1 \geq \dots \geq \lambda_N$ [11]. The matrix X can be computed efficiently based on the QR-decomposition as follows:

Let $H_w^T = QR$ be the QR-decomposition of H_w^T , where H_w is defined in Eq. (1), $Q \in \mathbb{R}^{n \times N}$ has orthonormal columns and $R \in \mathbb{R}^{N \times N}$ is upper triangular and nonsingular. Then $S_w = H_w H_w^T = R^T R$ and $(R^{-1})^T S_w R^{-1} = I_N$. That is, R^{-1} diagonalizes the within-class scatter matrix S_w . Next, consider the matrix

$$(R^{-1})^T S_b R^{-1} = (H_b^T R^{-1})^T (H_b^T R^{-1}) \equiv Y^T Y,$$

where $Y = H_b^T R^{-1}$.

Let $Y = U \Sigma V^T$ be the SVD of Y , where $U \in \mathbb{R}^{n \times q}$, $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_q) \in \mathbb{R}^{q \times q}$, $V \in \mathbb{R}^{N \times q}$, $\sigma_1 \geq \dots \geq \sigma_q$, and $q = \text{rank}(H_b)$. It is easy to check that $X = R^{-1}V$ diagonalizes both S_w and S_b and satisfies the conditions in Eq. (9) and Eq. (10).

It can be shown that the matrix consisting of the first q columns of X computed above (with normalization) solves the optimization problem in Eq. (7), where q is the rank of the matrix S_b , as stated in the following theorem:

Theorem 4.1. *Let the matrix X be defined as in Eq. (9) and Eq. (10), and $q = \text{rank}(S_b)$. Let $G^* = [\tilde{x}_1, \dots, \tilde{x}_q]$, where $\tilde{x}_i = \frac{1}{\sqrt{1+\lambda_i}} x_i$, x_i is the i -th column of the matrix X , and λ_i 's are defined in Eq. (10). Then G^* solves the optimization problem in Eq. (7).*

Proof. It is clear that the constraint in Eq. (7) is satisfied for $G = G^*$. Next we only need to show

that the maximum of $F(G)$ is obtained at G^* . By Eq. (9) and Eq. (10), we have

$$G^T S_w G = G^T X^{-T} (X^T S_w X) X^{-1} G = \tilde{G} \tilde{G}^T,$$

$$G^T S_b G = G^T X^{-T} (X^T S_b X) X^{-1} G = \tilde{G} \Lambda \tilde{G}^T,$$

where $\tilde{G} = (X^{-1}G)^T$. Hence, $F(G) = \text{trace} \left(\left(\tilde{G} \tilde{G}^T \right)^{-1} \left(\tilde{G} \Lambda \tilde{G}^T \right) \right)$. Let $\tilde{G}^T = QR$ be the QR-decomposition of $\tilde{G}^T \in \mathbb{R}^{N \times \ell}$ (note that \tilde{G}^T has full column rank), where $Q \in \mathbb{R}^{N \times \ell}$ has orthonormal columns and R is nonsingular. Using the fact that $\text{trace}(AB) = \text{trace}(BA)$, for any matrices A and B , we have

$$F(G) = \text{trace} \left((R^T R)^{-1} (R^T Q^T \Lambda Q R) \right) = \text{trace} (Q^T \Lambda Q) \leq \lambda_1 + \cdots + \lambda_q,$$

where the inequality becomes an equality for

$$Q = \begin{pmatrix} I_\ell \\ 0 \end{pmatrix} \text{ or } G = X \begin{pmatrix} I_\ell \\ 0 \end{pmatrix} R,$$

when the reduced dimension $\ell = q$. Note that R is an arbitrary upper triangular and nonsingular matrix. Hence, G^* corresponds to the case when R is set to be

$$R = \text{diag} \left(\frac{1}{\sqrt{1 + \lambda_1}}, \cdots, \frac{1}{\sqrt{1 + \lambda_q}} \right).$$

□

We are now ready to present our main result for this section:

Theorem 4.2. *Let \tilde{x}_i be defined as in Theorem 4.1. Then $\{\tilde{x}_i\}_{i=1}^q$ forms a set of optimal discriminant vectors for ULDA.*

Proof. By induction. It is trivial to check that $\tilde{x}_1 = \arg \max_{\phi} f(\phi)$, i.e., $\phi_1 = \tilde{x}_1$. Next assume $\phi_i = \tilde{x}_i$, for $i = 1, \dots, r$. We show in the following that $\phi_{r+1} = \tilde{x}_{r+1}$.

By the definition, $\phi_{r+1} = \arg \max_{\phi} f(\phi)$, subject to $\phi_{r+1}^T S \phi_i = 0$, for $i = 1, \dots, r$. Let $\phi_{r+1} = \sum_{i=1}^N \gamma_i \tilde{x}_i$, since $\{\tilde{x}_i\}_{i=1}^N$ forms a base for \mathbb{R}^N . By the constraints $\phi_{r+1}^T S \phi_i = 0$, for $i = 1, \dots, r$, we have $\gamma_i = 0$, for $i = 1, \dots, r$, hence $\phi_{r+1} = \sum_{i=r+1}^N \gamma_i \tilde{x}_i$. It follows from Eq. (9) and Eq. (10) that

$$\begin{aligned} f(\phi_{r+1}) &= \frac{\left(\sum_{i=r+1}^N \gamma_i \tilde{x}_i^T\right) S_b \left(\sum_{i=r+1}^N \gamma_i \tilde{x}_i\right)}{\left(\sum_{i=r+1}^N \gamma_i \tilde{x}_i^T\right) S_w \left(\sum_{i=r+1}^N \gamma_i \tilde{x}_i\right)} \\ &= \frac{\sum_{i=r+1}^N \gamma_i^2 \lambda_i}{\sum_{i=r+1}^N \gamma_i^2} \leq \frac{\sum_{i=r+1}^N \gamma_i^2 \lambda_{r+1}}{\sum_{i=r+1}^m \gamma_i^2} \\ &= \lambda_{r+1}, \end{aligned}$$

where the inequality becomes an equality if $\gamma_i = 0$, for $i = r+2, \dots, N$. Hence \tilde{x}_{r+1} can be chosen as the $(r+1)$ -th discriminant vector of ULDA, i.e., $\phi_{r+1} = \tilde{x}_{r+1}$. \square

An efficient algorithm for computing $\{\tilde{x}_i\}_{i=1}^q$ through QR-decomposition is presented below as

Algorithm 1.

5 The ULDA/GSVD algorithm

In the previous section, a variant of classical LDA criterion was presented in Eq. (7). It was shown that the solution to the optimization problem in Eq. (7) forms optimal discriminant vectors for classical ULDA. Thus, it provides an efficient way to compute optimal discriminant vectors for ULDA. However, the algorithm assumes the non-singularity of S_w , which limits its applicability to low-dimensional data. In [17], a subspace based method is presented to overcome the singularity problem, where the ULDA algorithm is preceded by PCA. However, the PCA stage may lose

Algorithm 1: The ULDA/QR Algorithm

Input: Data matrix A .

Output: Discriminant vectors \tilde{x}_i 's of ULDA.

1. Construct matrices H_w and H_b as in Eq. (1) and Eq. (2).
 2. Compute QR-decomposition on H_w^T as $H_w^T = QR$, where $Q \in \mathbb{R}^{n \times N}$, $R \in \mathbb{R}^{N \times N}$.
 3. Form the matrix $Y \leftarrow H_b^T R^{-1}$.
 4. Compute SVD on Y as $Y = U\Sigma V^T$, where $U \in \mathbb{R}^{n \times q}$, $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_q) \in \mathbb{R}^{q \times q}$,
 $V \in \mathbb{R}^{N \times q}$, $\sigma_1 \geq \dots \geq \sigma_q$, and $q = \text{rank}(H_b)$.
 5. $[x_1, \dots, x_q] \leftarrow R^{-1}V$.
 6. $\lambda_i \leftarrow \sigma_i^2$, for $i = 1, \dots, q$.
 7. $\tilde{x}_i \leftarrow \frac{1}{\sqrt{1+\lambda_i}}x_i$, for $i = 1, \dots, q$.
-

some useful information. In this section, we propose a new feature reduction algorithm, called ULDA/GSVD. The new criterion underlying ULDA/GSVD is motivated by the criterion in Eq. (7) and the regularized LDA. The new optimization problem for ULDA/GSVD is defined as follows,

$$G_\mu = \arg \max_{G^T S G = I_\ell} F_\mu(G), \quad (11)$$

where $F_\mu(G) = \text{trace}((G^T S_w G + \mu I_\ell)^{-1} G^T S_b G)$. Note that matrix $G^T S_w G + \mu I_\ell$ is guaranteed to be nonsingular for $\mu > 0$.

Recall that a limitation of regularized LDA is that the optimal value of the perturbation μ is difficult to determine. A key difference between ULDA/GSVD and regularized LDA is that the optimal solution to ULDA/GSVD is independent of the regularization parameter, i.e., $G_{\mu_1} = G_{\mu_2}$, for any $\mu_1, \mu_2 > 0$. The main result of this section is summarized in the following theorem:

Theorem 5.1. *Let G_μ^* , for any $\mu > 0$, be the optimal solution to the optimization problem in Eq. (11). Then the following equality holds:*

$$G_{\mu_1}^* = G_{\mu_2}^*, \text{ for any } \mu_1, \mu_2 > 0.$$

To prove Theorem 5.1, we first show how to compute G_μ^* , for any $\mu > 0$. Recall that when the within-class scatter matrix is nonsingular, the optimal transformation can be computed by finding the matrix X , which simultaneously diagonalizes the scatter matrices. For this, Generalized Singular Value Decomposition (GSVD) can be applied, even when both matrices are singular. A simple algorithm to compute GSVD can be found in [15], where the algorithm is based on [21].

The computation of G_μ^* , for any $\mu > 0$ is based on the following two lemmas.

Lemma 5.1. *Let S_w , S_b , and S be defined as in Section 2, and let $t = \text{rank}(S)$. Then there exists a nonsingular matrix $X \in \mathbb{R}^{N \times N}$, such that*

$$X^T S_b X = D_1 = \text{diag}(\alpha_1^2, \dots, \alpha_t^2, 0, \dots, 0),$$

$$X^T S_w X = D_2 = \text{diag}(\beta_1^2, \dots, \beta_t^2, 0, \dots, 0),$$

where $1 \geq \alpha_1 \geq \dots \geq \alpha_q > 0 = \alpha_{q+1} = \dots = \alpha_t$, $0 \leq \beta_1 \leq \dots \leq \beta_t \leq 1$, $D_1 + D_2 = \begin{pmatrix} I_t & 0 \\ 0 & 0 \end{pmatrix}$,

and $q = \text{rank}(S_b)$.

Proof. Let $K = \begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix}$, which is an $(n+k) \times N$ matrix. By the generalized singular value decomposition [21], there exist orthogonal matrices $U \in \mathbb{R}^{k \times k}$, $V \in \mathbb{R}^{n \times n}$, and a nonsingular matrix $X \in \mathbb{R}^{N \times N}$, such that

$$\begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}^T K X = \begin{bmatrix} \Sigma_1 & 0 \\ \Sigma_2 & 0 \end{bmatrix}, \quad (12)$$

where

$$\Sigma_1^T \Sigma_1 = \text{diag}(\alpha_1^2, \dots, \alpha_t^2), \quad \Sigma_2^T \Sigma_2 = \text{diag}(\beta_1^2, \dots, \beta_t^2),$$

$$1 \geq \alpha_1 \geq \dots \geq \alpha_q > 0 = \alpha_{q+1} = \dots = \alpha_t, \quad 0 \leq \beta_1 \leq \dots \leq \beta_t \leq 1,$$

$\alpha_i^2 + \beta_i^2 = 1$, for $i = 1, \dots, t$, and $q = \text{rank}(H_b) = \text{rank}(S_b)$.

Hence, $H_b^T X = U \begin{bmatrix} \Sigma_1 & 0 \end{bmatrix}$, and $H_w^T X = V \begin{bmatrix} \Sigma_2 & 0 \end{bmatrix}$. It follows that

$$X^T S_b X = X^T H_b H_b^T X = \begin{bmatrix} \Sigma_1^T \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} = D_1,$$

$$X^T S_w X = X^T H_w H_w^T X = \begin{bmatrix} \Sigma_2^T \Sigma_2 & 0 \\ 0 & 0 \end{bmatrix} = D_2,$$

$$\text{where } D_1 + D_2 = \begin{pmatrix} I_t & 0 \\ 0 & 0 \end{pmatrix}.$$

□

Lemma 5.2. *Define a trace optimization problem as follows:*

$$G = \arg \max_{G^T G = I_\ell} \text{trace} \left((G^T W G)^{-1} G^T B G \right), \quad (13)$$

where $W = \text{diag}(w_1, \dots, w_u) \in \mathbb{R}^{u \times u}$ is a diagonal matrix with $0 < w_1 \leq \dots \leq w_u$, and $B = \text{diag}(b_1, \dots, b_u) \in \mathbb{R}^{u \times u}$ is also diagonal with $b_1 \geq \dots \geq b_q > 0 = b_{q+1} = \dots = b_u$. Then $G^* = \begin{pmatrix} I_q \\ 0 \end{pmatrix}$ solves the optimization problem in Eq. (13) with $\ell = q$.

Proof. It is clear that the constraint in the optimization in Eq. (13) is satisfied for G^* with $\ell = q$.

Next, we show that G^* solves the following optimization problem:

$$G = \arg \max_G \text{trace} \left((G^T W G)^{-1} G^T B G \right). \quad (14)$$

It is well known that the solution can be obtained by solving the eigenvalue problem on $W^{-1}B$, since W is nonsingular. Note that $W^{-1}B$ is diagonal and only the first q diagonal entries are nonzero. Hence e_i , for $i = 1, \dots, q$, is the eigenvector of $W^{-1}B$ corresponding to the i -th largest eigenvalue, where $e_i = (0, \dots, 1, 0, \dots, 0)^T$ and the 1 appears at the i -th position. Therefore $G^* = \begin{pmatrix} I_q \\ 0 \end{pmatrix}$ solves the optimization in Eq. (14). □

With Lemma 5.1 and Lemma 5.2, we can compute G_μ^* , for any $\mu > 0$ as follows:

Theorem 5.2. *Let the matrix X be defined as in Lemma 5.1, and let $q = \text{rank}(S_b)$. Then*

$$G_\mu^* = X \begin{pmatrix} I_q \\ 0 \end{pmatrix} \text{ solves the optimization problem in Eq. (11) with } \ell = q.$$

Proof. By Lemma 5.1, $X^T S_b X = D_1$, $X^T S_w X = D_2$, where the two diagonal matrices D_1 and D_2 satisfy $D_1 + D_2 = \begin{pmatrix} I_t & 0 \\ 0 & 0 \end{pmatrix}$. It is easy to check that

$$\begin{aligned} (G_\mu^*)^T S G_\mu^* &= (I_q, 0) X^T (S_b + S_w) X \begin{pmatrix} I_q \\ 0 \end{pmatrix} \\ &= (I_q, 0) (D_1 + D_2) \begin{pmatrix} I_q \\ 0 \end{pmatrix} = I_q, \end{aligned}$$

i.e., the constraint in the optimization problem in Eq. (11) is satisfied. Next we show G_μ^* minimizes $F_\mu(G)$.

Since

$$\begin{aligned} G^T S_b G &= G^T (X^{-1})^T (X^T S_b X) X^{-1} G = \tilde{G} D_1 \tilde{G}^T, \\ G^T S_w G &= G^T (X^{-1})^T (X^T S_w X) X^{-1} G = \tilde{G} D_2 \tilde{G}^T, \end{aligned}$$

where $\tilde{G} = (X^{-1} G)^T$, $F_\mu(G)$ can then be rewritten as,

$$F_\mu(G) = \text{trace} \left(\left(\tilde{G} D_2 \tilde{G}^T + \mu I_\ell \right)^{-1} \tilde{G} D_1 \tilde{G}^T \right). \quad (15)$$

Let $\tilde{G} = \begin{pmatrix} G_1^T & G_2^T \end{pmatrix}$ be a partition of \tilde{G} , such that $G_1^T \in \mathbb{R}^{\ell \times t}$ and $G_2^T \in \mathbb{R}^{\ell \times (N-t)}$.

By the constraint that $G^T S G = I_\ell$, we have

$$\begin{aligned} I_\ell &= G^T S G = G^T (S_w + S_b) G = G^T S_b G + G^T S_w G \\ &= \tilde{G} D_1 \tilde{G}^T + \tilde{G} D_2 \tilde{G}^T = \tilde{G} (D_1 + D_2) \tilde{G}^T = G_1^T G_1. \end{aligned}$$

Algorithm 2: The ULDA/GSVD Algorithm

Input: Data matrix A

Output: Optimal transformation matrix G^*

1. Form H_b and H_w as in Eq. (2) and Eq. (1).
 2. Compute GSVD on the matrix pair (H_b^T, H_w^T) to obtain the matrix X , as in Lemma 5.1.
 3. $q \leftarrow \text{rank}(H_b)$.
 4. $G^* \leftarrow [X_1, \dots, X_q]$.
-

Hence, $F_\mu(G)$ in Eq. (15) can be rewritten as

$$F_\mu(G) = \text{trace} \left((G_1^T (D_2^t + \mu I_\ell) G_1)^{-1} G_1^T D_1^t G_1 \right).$$

where D_1^t and D_2^t are the t -th leading sub-matrices of D_1 and D_2 respectively. It is clear that $F_\mu(G)$ is independent of G_2 . Hence we can simplify set $G_2 = 0$. Denote $\Sigma = (D_2^t + \mu I_t)$, which is a nonsingular and diagonal matrix. It follows that

$$F_\mu(G) = \text{trace} \left((G_1^T \Sigma G_1)^{-1} G_1^T D_1^t G_1 \right).$$

The result then follows from Lemma 5.2, with $W = \Sigma$ and $B = D_1^t$. \square

Theorem 5.2 implies that the optimal solution G_μ^* to the optimization in Eq. (11) only depends on X , which is determined by H_w and H_b , hence it is independent of μ . That is, $G_{\mu_1}^* = G_{\mu_2}^*$, for any $\mu_1, \mu_2 > 0$. This completes the proof of the main result of this section summarized in Theorem 5.1.

The computation of the optimal transformation G^* is summarized in **Algorithm 2**.

5.1 Efficient computation of diagonalizing matrix X

In Lemma 5.1, a nonsingular matrix X is computed by applying GSVD, which can be expensive, especially for large matrices. A key property of X which leads to the optimal solution G^* is that it diagonalizes the scatter matrices simultaneously. In this section, we take a closer look at the relationship among the three scatter matrices S_b , S_w , and S , and present an efficient algorithm for diagonalizing them simultaneously without using GSVD.

Let $H_t = U\Sigma V^T$ be the SVD of H_t , where H_t is defined in Eq. (3). Then

$$S = H_t H_t^T = U\Sigma V^T V \Sigma^T U^T = U\Sigma \Sigma^T U^T.$$

That is, the eigen-decomposition of S can be obtained by computing the SVD of H_t .

Let $U = (U_1, U_2)$, be the partition of U , such that $U_1 \in \mathbb{R}^{m \times t}$ and $U_2 \in \mathbb{R}^{m \times (m-t)}$, where $t = \text{rank}(S)$.

Let $\Sigma \Sigma^T = \begin{pmatrix} \Sigma_t^2 & 0 \\ 0 & 0 \end{pmatrix}$, where $\Sigma_t \in \mathbb{R}^{t \times t}$ is diagonal and non-singular.

Since $S = S_b + S_w$, we have

$$\begin{aligned} \begin{pmatrix} \Sigma_t^2 & 0 \\ 0 & 0 \end{pmatrix} &= \begin{pmatrix} U_1^T \\ U_2^T \end{pmatrix} S_b(U_1, U_2) + \begin{pmatrix} U_1^T \\ U_2^T \end{pmatrix} S_w(U_1, U_2) \\ &= \begin{pmatrix} U_1^T S_b U_1 & U_1^T S_b U_2 \\ U_2^T S_b U_1 & U_2^T S_b U_2 \end{pmatrix} + \begin{pmatrix} U_1^T S_w U_1 & U_1^T S_w U_2 \\ U_2^T S_w U_1 & U_2^T S_w U_2 \end{pmatrix} \end{aligned} \quad (16)$$

It follows that $U_2^T S_b U_2 + U_2^T S_w U_2 = 0$, since the second diagonal block of the matrix on the left hand side of the equation is zero. Therefore, $U_2^T S_b U_2 = 0$ and $U_2^T S_w U_2 = 0$, since both are positive semi-definite. We thus have $U_1^T S_b U_2 = 0$ and $U_1^T S_w U_2 = 0$, since both matrices on the right hand

side of Eq. (16) are positive semi-definite. That is,

$$U^T S_b U = \begin{pmatrix} U_1^T S_b U_1 & 0 \\ 0 & 0 \end{pmatrix}, \quad U^T S_w U = \begin{pmatrix} U_1^T S_w U_1 & 0 \\ 0 & 0 \end{pmatrix}. \quad (17)$$

From Eq. (16) and Eq. (17), we have

$$\Sigma_t^2 = U_1^T S_b U_1 + U_1^T S_w U_1. \quad (18)$$

By multiplying both sides of Eq. (18) by Σ_t^{-1} from the left and the right, we have

$$I_t = \Sigma_t^{-1} U_1^T S_b U_1 \Sigma_t^{-1} + \Sigma_t^{-1} U_1^T S_w U_1 \Sigma_t^{-1}. \quad (19)$$

Next, we diagonalize the matrix $\Sigma_t^{-1} U_1^T S_b U_1 \Sigma_t^{-1}$ in Eq. (19).

Denote $B = \Sigma_t^{-1} U_1^T H_b$ and let $B = P \tilde{\Sigma} Q^T$ be the SVD of B , where P and Q are orthogonal and $\tilde{\Sigma}$ is diagonal. Then $\Sigma_t^{-1} U_1^T S_b U_1 \Sigma_t^{-1} = P \tilde{\Sigma}^2 P^T = P \Sigma_b P^T$, where $\Sigma_b = \tilde{\Sigma}^2 = \text{diag}(\lambda_1, \dots, \lambda_t)$, $\lambda_1 \geq \dots \geq \lambda_q > 0 = \lambda_{q+1} = \dots = \lambda_t$, and $q = \text{rank}(S_b)$.

It follows from Eq. (19) that $I_t = \tilde{\Sigma}^2 + P^T \Sigma_t^{-1} U_1^T S_w U_1 \Sigma_t^{-1} P$. That is, the matrix P diagonalizes the matrix $\Sigma_t^{-1} U_1^T S_b U_1 \Sigma_t^{-1}$ in Eq. (19). It is clear that $P^T \Sigma_t^{-1} U_1^T S_w U_1 \Sigma_t^{-1} P = I_t - \tilde{\Sigma}^2 = \Sigma_w$ is also diagonal.

Combining all these together, we have

$$X^T S_b X = \begin{pmatrix} \Sigma_b & 0 \\ 0 & 0 \end{pmatrix} \equiv D_b, \quad X^T S_w X = \begin{pmatrix} \Sigma_w & 0 \\ 0 & 0 \end{pmatrix} \equiv D_w, \quad X^T S X = \begin{pmatrix} I_t & 0 \\ 0 & 0 \end{pmatrix} \equiv D_t, \quad (20)$$

where

$$X = U \begin{pmatrix} \Sigma_t^{-1} P & 0 \\ 0 & I \end{pmatrix}. \quad (21)$$

Thus the matrix X in Eq. (21) simultaneously diagonalizes S_b , S_w , and S . The pseudo-code for the simultaneous diagonalization is given in **Algorithm 3**.

Algorithm 3: Efficient computation of diagonalizing matrix X

Input: data matrix A

Output: matrix X

1. Form three matrices H_b and H_t as in Eqs. (2)–(3).
 2. Compute SVD of H_t as $H_t = U_1 \Sigma_t V_1^T$.
 4. $B \leftarrow \Sigma_t U_1^T H_b$.
 5. Compute SVD of B as $B = P \Sigma Q^T$; $q \leftarrow \text{rank}(B)$.
 6. $X \leftarrow U \begin{pmatrix} \Sigma_t P & 0 \\ 0 & I \end{pmatrix}$.
-

5.2 Relationship between ULDA/GSVD and ULDA/QR

In this section, we show that ULDA/GSVD is equivalent to ULDA/QR when the within-class scatter matrix S_w is nonsingular. Therefore, ULDA/QR can be considered as a special case of ULDA/GSVD when S_w is nonsingular. Note that ULDA/GSVD is more general in the sense that it is applicable regardless of the singularity of S_w .

Recall that ULDA/QR involves the matrix X , which satisfies

$$\begin{aligned} X^T S_w X &= I_N, \\ X^T S_b X &= \Lambda = \text{diag}(\lambda_1, \dots, \lambda_N), \end{aligned}$$

where $\lambda_1 \geq \dots \geq \lambda_N$.

The final transformation matrix $G^* = [\tilde{x}_1, \dots, \tilde{x}_q]$, where $\tilde{x}_i = \frac{1}{\sqrt{1+\lambda_i}} x_i$, x_i is the i -th column of

the matrix X . It follows that

$$(G^*)^T S G^* = I_q, \quad (22)$$

$$(G^*)^T S_b G^* = \text{diag} \left(\frac{\lambda_1}{1 + \lambda_1}, \dots, \frac{\lambda_q}{1 + \lambda_q} \right), \quad (23)$$

Since $f(x) = x/(1 + x)$ is an increasing function, we have

$$\frac{\lambda_1}{1 + \lambda_1} \geq \dots \geq \frac{\lambda_q}{1 + \lambda_q}.$$

Thus the transformation matrix G^* from ULDA/QR satisfies the conditions in Lemma 5.1 for ULDA/GSVD. That is, ULDA/GSVD is equivalent to ULDA/QR, when the within-class scatter matrix S_w is nonsingular. Note that ULDA/QR is not applicable when S_w is singular. ULDA/GSVD can thus be considered as an extension of ULDA/QR for a singular within-class scatter matrix. In the following experimental studies, we only focus on the ULDA/GSVD algorithm.

6 Experiments

We evaluate the effectiveness of the ULDA/GSVD algorithm in this section, which consists of three parts. The first part describes our test datasets. The second part examines the effect of the number of reduced dimensions on the classification performance of ULDA/GSVD. In the last part, we compare our ULDA/GSVD algorithm with PCA, OCM, and subspace ULDA, in terms of classification accuracy. The K-Nearest-Neighbor (K-NN) algorithm with different K is used as our classifier.

6.1 Datasets

We used two datasets: Spambase and Wine from the UCI Machine Learning Repository³. We used a subset of the original Spambase dataset, which consists of spam and non-spam emails. Most of the features indicate whether a particular word or character occurred frequently in the e-mail. The Wine dataset is the result of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The features correspond to the quantities of 13 different constituents found in each of the three types of wines. For these two datasets, the data dimension (N) is much smaller than the sample size (n). We also used six other datasets: GCM, ALL, tr41, re1, PIX, and ORL, where the data dimension is much larger than the sample size. In this case, ULDA/QR is not applicable, since all scatter matrices are singular, while ULDA/GSVD is still applicable. GCM [25, 31] and ALL [32] are microarray gene expression datasets; tr41 is a document dataset derived from the TREC-5, TREC-6, and TREC-7 collections⁴; re1 is another document dataset derived from *Reuters-21578* text categorization test collection Distribution 1.0⁵; and ORL⁶ and PIX⁷ are two face image datasets.

Table 2 summarizes the statistics of our test datasets.

6.2 Effect of the number of reduced dimensions on ULDA/GSVD

In this experiment, we study the effect of the number of reduced dimensions on the classification performance of ULDA/GSVD. The number of reduced dimensions ranges from 1 to 20. The

³<http://www.ics.uci.edu/~mlearn/MLRepository.html>

⁴<http://trec.nist.gov>

⁵<http://www.research.att.com/~lewis>

⁶<http://www.uk.research.att.com/facedatabase.html>

⁷<http://peipa.essex.ac.uk/ipa/pix/faces/manchester/test-hard/>

Table 2. Statistics for the test datasets ('—' means that the natural splitting of the dataset into training and test set is not available. For Spambase, Wine, GCM, and ALL, the original training and test sets are given, while for tr41, re1, PIX, and ORL, the original splitting is not provided.)

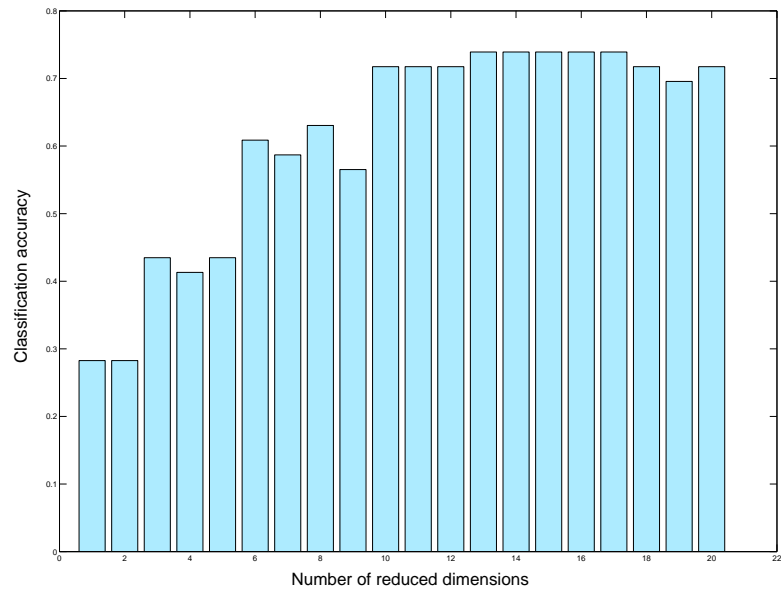
Dataset	Size (n)			Dimension (N)	Number of classes (k)
	training	test	total		
Spambase	400	600	1000	57	2
Wine	118	60	178	13	3
GCM	144	46	190	11485	14
ALL	163	85	248	12559	6
tr41	—	—	210	7454	7
re1	—	—	490	3759	5
PIX	—	—	300	10000	30
ORL	—	—	400	10304	40

classification results on the GCM and ALL datasets are shown in Figure 1, where the horizontal axis is the number of reduced dimensions and the vertical axis is the classification accuracy. We can observe that the accuracy tends to increase when the number of reduced dimensions increases, until $q = \text{rank}(H_b)$ (13 for GCM and 5 for ALL) is reached. Similar trends have been observed from other datasets, and the results are not presented. In the following experiment, we set the reduced dimension to be $\text{rank}(H_b)$ for ULDA/GSVD.

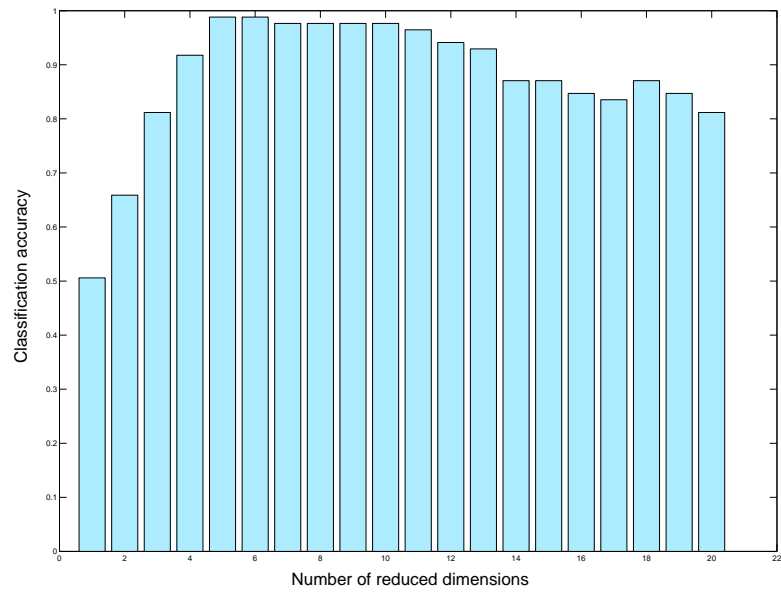
6.3 Comparison of classification accuracy

In this experiment, we applied ULDA/GSVD to the eight datasets from Table 2 and compared with OCM, PCA, and subspace ULDA in term of classification accuracy. The results are summarized in Table 3. The number of principal components used in PCA and Subspace ULDA is determined through cross-validation, and may be different for different datasets.

For datasets, including Spambase, Wine, GCM, and ALL, the training and test sets given in the original datasets are used for computing the accuracy. For the other four datasets, including tr41,



GCM



ALL

Figure 1. Effect of the number of reduced dimensions on the classification performance of ULDA/GSVD for the GCM (top) and ALL (bottom) datasets. The optimal numbers of reduced dimensions for GCM and ALL are 13 and 5, respectively.

re1, PIX, and ORL, where the training and test sets are not given, we performed our study by repeated random splitting into training and test sets exactly as in [7]. The data was partitioned randomly into a training set consisting of two-thirds of the whole set and a test set consisting of one-third of the whole set. To reduce the variability, the splitting was repeated 50 times and the resulting accuracies were averaged. The standard deviation for each dataset was also reported.

The main observations from Table 3 include:

- ULDA/GSVD is competitive with the other three algorithms for all datasets in terms of classification. Subspace ULDA performs well for most datasets. However, subspace ULDA applies cross-validation for determining the optimal set of principal components in the PCA step, which can be expensive, especially for large datasets.
- ULDA/GSVD is extremely stable under different K-NN classifiers for all datasets, whereas the performance of OCM and PCA degrades for many cases, as the number, K , of nearest neighbors increases. Subspace ULDA is also stable under different K-NN classifiers for most datasets.
- PCA does not perform in many cases. This is likely related to the fact that PCA is unsupervised and does not use the class label information, while the other three algorithms fully utilize the class label information. OCM performs well for the two document datasets and the two face image datasets, while it performs poorly for the other datasets.

Table 3. Comparison of classification accuracy on four different methods. For tr41, re1, PIX, and ORL, the mean accuracy and standard deviation (in parenthesis) for 50 different runs are reported.

Dataset	Method	1NN	3NN	5NN	10NN
Spambase	ULDA	88.17%	89.00%	89.67%	89.50%
	Subspace	85.67%	87.67%	85.67%	85.50%
	OCM	59.83%	59.00%	65.17%	69.83%
	PCA	66.50%	64.00%	67.83%	71.00%
Wine	ULDA	96.67%	96.67%	98.33%	98.33%
	Subspace	96.67%	96.67%	98.33%	95.00%
	OCM	68.33%	75.00%	68.33%	65.00%
	PCA	70.00%	75.00%	68.33%	65.00%
GCM	ULDA	73.91%	73.91%	73.91%	73.91%
	Subspace	73.91%	67.39%	65.22%	71.74%
	OCM	58.70%	56.52%	52.17%	47.83%
	PCA	60.87%	56.52%	43.48%	43.48%
ALL	ULDA	98.82%	98.82%	98.82%	98.82%
	Subspace	95.29%	95.29%	95.29%	95.29%
	OCM	95.29%	95.29%	95.29%	95.29%
	PCA	96.47%	95.29%	95.29%	95.29%
tr41	ULDA	97.74% (1.47)	98.09% (1.46)	97.63% (1.72)	97.74% (1.92)
	Subspace	95.20% (2.50)	96.54% (2.06)	96.40% (1.78)	96.74% (2.14)
	OCM	96.14% (2.26)	96.34% (2.47)	95.57% (2.07)	95.94% (2.33)
	PCA	87.37% (2.82)	84.06% (4.73)	82.94% (3.87)	81.00% (4.64)
re1	ULDA	94.97% (1.62)	94.92% (1.54)	94.72% (1.73)	94.96% (1.43)
	Subspace	94.03% (1.70)	94.52% (1.37)	94.75% (1.85)	94.86% (1.54)
	OCM	93.19% (1.88)	94.31% (1.45)	94.36% (2.07)	94.60% (1.33)
	PCA	87.90% (2.57)	88.84% (2.17)	89.71% (2.37)	90.67% (2.32)
PIX	ULDA	96.76% (1.60)	96.64% (1.60)	96.47% (1.55)	96.71% (1.85)
	Subspace	95.84% (1.89)	95.87% (2.07)	96.04% (2.03)	95.40% (2.50)
	OCM	96.84% (1.76)	94.69% (1.97)	93.56% (2.30)	87.80% (2.66)
	PCA	97.16% (1.65)	93.78% (2.07)	91.78% (2.62)	83.51% (2.47)
ORL	ULDA	93.13% (2.00)	93.38%(2.08)	93.62% (2.41)	93.07% (2.06)
	Subspace	93.70% (2.25)	93.38%(1.94)	93.58% (2.30)	92.90% (2.21)
	OCM	96.57% (1.33)	93.45%(2.35)	90.70% (2.60)	82.08% (2.94)
	PCA	95.65% (1.51)	92.23%(2.32)	87.20% (2.52)	73.33% (2.79)

7 Conclusion

Uncorrelated features with minimum redundancy are highly desirable in feature reduction for many applications such as text retrieval, image retrieval, etc. In this paper, we presented a study on uncorrelated Linear Discriminant Analysis (ULDA). This study contains two major contributions. The first one is the theoretical result on the equivalence between classical ULDA and classical LDA, which leads to a fast implementation of ULDA, ULDA/QR. Then we propose ULDA/GSVD, based on a novel optimization criterion, that can successfully overcome the singularity problem in classical ULDA. The criterion used in ULDA/GSVD is the perturbed version of the one from ULDA/QR, while the solution to ULDA/GSVD is shown to be independent of the amount of perturbation applied, thus avoiding the limitation in regularized LDA. Experimental results on various types of data show the superiority of ULDA/GSVD over other competing algorithms including PCA, OCM, and subspace ULDA.

Experimental results show that ULDA/GSVD is extremely stable under different K-NN classifiers for all datasets. We plan to carry out detailed theoretical analysis on this in the future. The current work focuses on linear discriminant analysis, which applies a linear decision boundary. Discriminant analysis can also be studied in a non-linear fashion—so-called kernel discriminant analysis—, by using the kernel trick [26]. This is desirable if the data has weak linear separability. We plan to extend the current work to deal with the nonlinearity in the future.

Acknowledgment

We thank the the four reviewers and the Associate Editor for their comments, which helped improve the paper significantly.

Research of J. Ye and R. Janardan was sponsored, in part, by the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement number DAAD19-01-2-0014, the content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. Research of J. Ye was also sponsored, in part, by the Evolutionary Functional Genomics Center of the Biodesign Institute at Arizona State University. Fellowships from Guidant Corporation and from the Department of Computer Science & Engineering, at the University of Minnesota, Twin Cities is gratefully acknowledged.

The work of H. Park has been performed while serving as a program director at the National Science Foundation (NSF) and was partly supported by IR/D from the NSF. Her work was also supported in part by the National Science Foundation Grants CCR-0204109 and ACI-0305543. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- [1] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
- [2] M. Berry, S. Dumais, and G. O’Brie. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37:573–595, 1995.
- [3] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33:1713–1726, 2000.
- [4] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Scienc*, 41:391–407, 1990.
- [5] L. Duchene and S. Leclerq. An optimal transformation for discriminant and principal component analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(6):978–983, 1988.
- [6] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, 2000.
- [7] S. Dudoit, J. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002.
- [8] D. Foley and J. Sammon. An optimal set of discriminant vectors. *IEEE Trans. Computers*, 24(3):281–289, 1975.
- [9] J. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.
- [10] K. Fukunaga. *Introduction to Statistical Pattern Classification*. Academic Press, San Diego, California, USA, 1990.
- [11] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, USA, third edition, 1996.

- [12] T. Hastie, A. Buja, and R. Tibshirani. Penalized discriminant analysis. *Annals of Statistics*, 23:73–102, 1995.
- [13] T. Hastie and R. Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society series B*, 58:158–176, 1996.
- [14] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.
- [15] P. Howland, M. Jeon, and H. Park. Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 25(1):165–179, 2003.
- [16] P. Howland and H. Park. Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(8):995–1006, 2004.
- [17] Z. Jin, J. Y. Yang, Z.-S. Hu, and Z. Lou. Face recognition based on the uncorrelated discriminant transformation. *Pattern Recognition*, 34:1405–1416, 2001.
- [18] Z. Jin, J.-Y. Yang, Z.-M. Tang, and Z.-S. Hu. A theorem on the uncorrelated optimal discriminant vectors. *Pattern Recognition*, 34(10):2041–2047, 2001.
- [19] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.
- [20] W. Krzanowski, P. Jonathan, W. McCarthy, and M. Thomas. Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data. *Applied Statistics*, 44:101–115, 1995.
- [21] C. Paige and M. Saunders. Towards a generalized singular value decomposition. *SIAM Journal on Numerical Analysis*, 18:398–405, 1981.
- [22] C. Park and H. Park. A comparison of generalized LDA algorithms for undersampled problems. In submission.

- [23] C. Park and H. Park. A relationship between LDA and the generalized minimum squared error solution. *SIAM Journal on Matrix Analysis and Applications*, to appear, 2005.
- [24] H. Park, M. Jeon, and J. Rosen. Lower dimensional representation of text data based on centroids and least squares. *BIT*, 43(2):1–22, 2003.
- [25] Ramaswamy, S. et al. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Science*, 98(26):15149–15154, 2001.
- [26] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.
- [27] D. L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(8):831–836, 1996.
- [28] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. Computer Vision and Pattern Recognition Conference*, pages 586–591, 1991.
- [29] J. Ye. Characterization of a family of algorithms for generalized discriminant analysis on under-sampled problems. *Journal of Machine Learning Research*, 6:483–502, 2005.
- [30] J. Ye, R. Janardan, C. Park, and H. Park. An optimization criterion for generalized discriminant analysis on undersampled problems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(8):982–994, 2004.
- [31] Yeang, C.H. et al. Molecular classification of multiple tumor types. *Bioinformatics*, 17(1):1–7, 2001.
- [32] Yeoh, E.J. et al. Classification, subtype discovery, and prediction of outcome in pediatric lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1(2):133–143, 2002.